Modelling of a Double Clutch Transmission with an Appropriate Controller for the Simulation of Shifting Processes

Henrik Isernhagen Clemens Gühmann
Technische Universität Berlin
Chair of Electronic Measurement and Diagnostic Technology
Sekr. EN 13, Einsteinufer 17, 10587 Berlin
{henrik.isernhagen, clemens.guehmann}@tu-berlin.de

Abstract

In this paper the modelling of a double clutch transmission with an appropriate controller is presented. An accordant library for modelling different levels of detail and the use of defined state signals are introduced. Furthermore, the control of the transmission with the simulation of shifting cycles is discussed. By varying the driver models it is possible to drive miscellaneous drive and shifting cycles. We present simulation results of a drive cycle with an examination of the interaction between the transmission control and the engine control. Finally, the application of the model and the simulation data are shown in view to the parameterisation of an automated measurement data analysis system.

Keywords: double clutch transmission, power train, control, shifting processes, state signals, simulation

1 Introduction

Nowadays, more and more new cars are assembled with double clutch transmission because of efficiency, drive comfort, and uninterrupted power shifts. A double clutch transmission contains two parallel transmission shafts with two parallel clutches. The clutches can be either dry clutches or laminar multi-disc (wet) clutches. The different gears are mounted alternately on the two transmission shafts. The first shaft contains the odd gears, and the second shaft contains the even gears. Depending on the number of gears the reverse gear is mounted either on the first shaft or on the second shaft. There are uninterrupted power shifts possible by reason of the two different clutches with the accordant shafts. The first clutch opens and the second clutch closes simultaneously during the shift process resulting in an uninterrupted power transmission. The correct control of the two clutches is very import in view to the different shifting processes. At this point it is necessary to distinguish between upshifts and downshifts as well as pulling power and pushing power of the power train. Control and calibration errors can result in bad shifting, e. g., in form of revolution speed droppings, break outs, or oscillations. These errors should be detected by an automated measurement data analysis system [1]. The software is used for the evaluation of power train measurement data in the vehicle development. The parameterisation of this analysis system requires measurement data of good and bad shifting processes. Accordingly, the aim of the modelling and the simulation are to get data of shifting processes of different quality.

In this paper we present the modelling, control, and simulation of double clutch transmissions. First, a basic library and the use of state signals are introduced in Section 2. In Section 3 we discuss the need for models with different levels of detail. Section 4 shows the control structure of the double clutch transmission in view to different shifting processes, and Section 5 describes the simulation of drive and shifting cycles. Finally, we discuss in Section 6 simulation results of particular shifting cycles, and we present an outlook to the parameterisation of an automated measurement data analysis system [1] using the simulation results.

2 Library for modelling double clutch transmissions

For the modelling of double clutch transmissions we developed a library with basic components. On the one hand there are auxiliaries, interfaces, and basic blocks, which are reused in further blocks and models. These blocks are often used and they normally have a simple structure. On the other hand the library contains different components of the double clutch transmission.

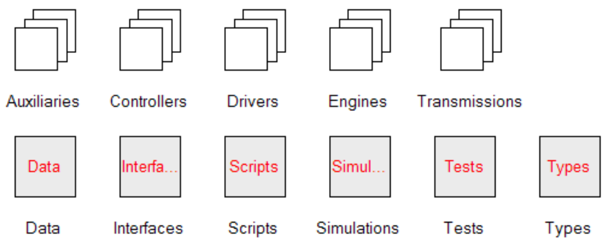

Figure 1: Library base structure

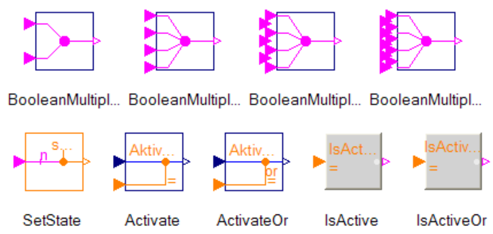

Figure 2: State signals relevant blocks

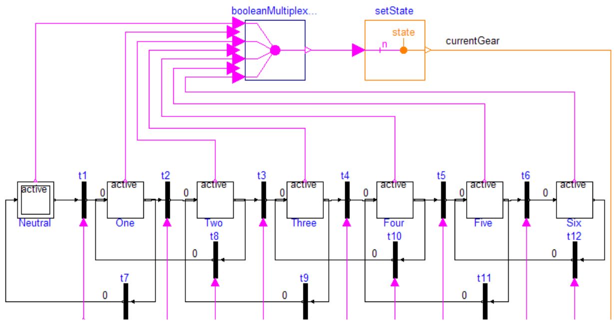

Figure 3: Generating a state signal in a state graph

Here, it has to be differentiated between physical components of the transmission and components for control. A couple of models exist with different levels of detail or with a different implementation, e. g., gearbox with or without losses, or different controls for shifting. Furthermore, the library contains additional components for modelling, control, and simulation of the power train with a double clutch transmission. An example is the engine control in connection with the transmission control. Here, the engine torque has to be influenced by the transmission control in some parts of the shifting process for a smooth shifting.

The library is derived from the *VehicleInterfaces* library [2] and the *PowerTrain* library [3]. Consequently, the suitable interfaces and bus structures are used, so that new transmission models can be connected with the models of these libraries. Furthermore, we defined additional interfaces to get the advantage of replaceable models and blocks. Doing this, it is possible to define some basic models and structures and it is easy to build models with different levels of detail. The overall structure of the library is arranged according to the different components like controllers, engines, transmissions, auxiliaries, interfaces, etc. The library base structure is shown in Figure 1. Besides, the *VehicleInterfaces* library, the *PowerTrain* library, and the Modelica Standard Library [4] are used.

2.1 Introduction of state signals

State graphs [5] are used at important places for the control of the double clutch transmission, because the control process of the double clutch transmission depends often on more than one state. The active states of the particular state graphs have to be available through the control from different points. According to this, it is necessary to get an access to the active states. Here, we introduced so called state signals which are accessible in our case by the *TransmissionControlBus* in the *ControlBus* of the *VehicleInterfaces*. A state signal is defined by an Integer and the particular states are represented by the present Modelica implementation of the Enumerations. Inside a state graph the particular states are combined to one state signal, which represents always the active state by the associated Integer value of the Enumeration. Additionally, we created new blocks to access these state signals so that the state signal can be compared with a desired state and a specific control can be activated.

The different blocks for using state signals are shown in Figure 2. The blocks in the first row and the block `SetState` are used to generate the desired state signal. Figure 3 shows an example for generating a state signal in a state graph. The input size of the `BooleanMultiplex` depends here on the number of states in the state graph. The block `SetState` determines the active state and assigns the accordant Enumeration value to the state signal. The code is shown in Listing 1.

There are two types of blocks to access the state signals. First, a real signal can pass or is set to zero by the blocks `Activate` and `ActivateOr`. These blocks work like a switch. Listing 2 shows the code example for the block `Activate`. Second, the blocks `IsActive` and `IsActiveOr` generate a Boolean signal according to the signal state. All four blocks are parameterised by the chosen Enumeration value. They compare the actual state signal and the Enumeration value and set the output signal accordingly. The 'or'-variants use a logical 'or' comparison to different Enumeration val-

Listing 1: Code example for `SetState`

```
model SetState
  extends
        Modelica.Blocks.Interfaces.IntegerSO;
  parameter Types.DCTTypes.Temp
        states[:]={1,2};
  Modelica.Blocks.Interfaces.BooleanInput
        u[size(states, 1)];
protected
  Integer sz =  size(states, 1);
algorithm
  for i in 0:sz-1 loop
    if u[sz-i] then
      y:=states[sz-i];
    end if;
  end for;
end SetState;
```

Listing 2: Code example for `Activate`

```
block Activate
  extends
        Modelica.Blocks.Interfaces.SO;
  parameter VehicleDCT.Types.DCTTypes.Temp
        state(start=1);
  Modelica.Blocks.Interfaces.RealInput u;
  Modelica.Blocks.Interfaces.IntegerInput
        statesignal;
equation
  y = if (statesignal==state)
    then u else 0;
end Activate;
```

ues, respectively.
Furthermore, an advantage is the possibility to plot the state signals like other simulation variables. Thus, the change of the states of each state graph can be observed after the simulation. A possible disadvantage is the increase of variables and equations.

3 Different models of a double clutch transmission

The aim of the simulation process is to get data of shift processes of different quality. According to this, it is necessary to build models with different levels of detail. A basic model of a double clutch transmission is shown in Figure 4. It represents the base for further models by containing replaceable elements in the front and back layers using the interfaces and elements of the described library.
Figure 4 shows the two shafts with the two clutches. The first shaft contains the odd gears and the reverse gear, and the second shaft contains the even gears. The

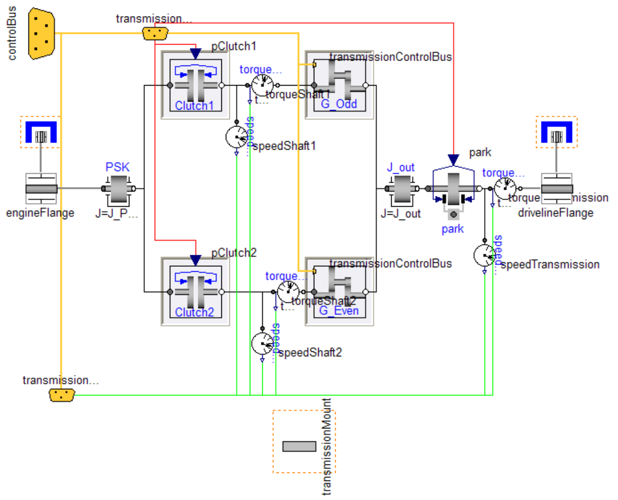

Figure 4: Basic double clutch transmission model

synchronisers of the particular gears are also modelled as clutches. Here, the transmission control has to ensure the correct gear synchronisations. Only one gear per shaft is allowed to be synchronised. The *TransmissionControlBus* contains the control signals for the two parallel clutches and for the particular synchronisers. All measured signals like revolution speeds and torques are added to the *TransmissionBus*. In a real vehicle torques are not measured, because of the difficult measurement of a dynamic torque. Normally, the torque is calculated by characteristic curves.
The replaceable elements can be modelled with different levels of detail. A clutch can be either a dry or a laminar clutch. Many components can be chosen as ideal or with losses. Furthermore, it is possible to use different numbers of spring and damper elements and different spring and damping constants.

4 Control of the transmission

The modelling of the control is just as important as the physical modelling of the double clutch transmission. The outputs of the transmission control are the control signals for the two clutches and for the synchronisers of the particular gears. Additionally, the transmission control calculates internal control signals like the actual driving state, the current gear, or the requested gear. The main focus is the control of the different shifting processes. At this point, the correct control is very important in view to smooth shiftings, because both clutches are engaged. Control errors can result in oscillations, jerks, or even in a damage of the transmission.
Figure 5 shows the structure of the transmission con-

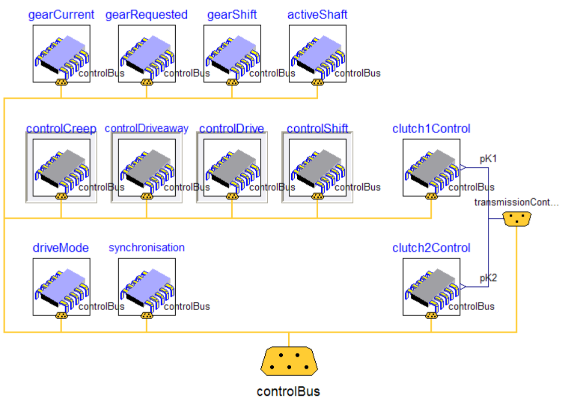

Figure 5: Transmission control model

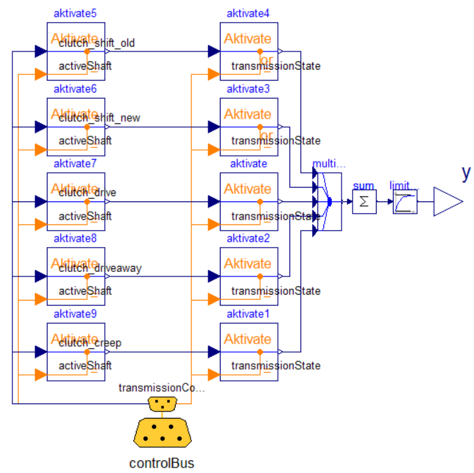

Figure 6: Clutch control model merging the particular clutch control values

trol. All control components extend the *Controllers* interface and use *ControlBus* sub buses of the *VehicleInterfaces*. The models `GearCurrent` and `GearRequested` determine the current and the requested gear using state graphs and assigning the states to state signals. Additionally, `DriveMode` appoints the actual driving state, e. g., creep, drive, upshift, etc., and assigns a corresponding state signal, too. The block `GearShift` identifies the type of the shifting using the drive mode and the power transmission direction of the power train. `ActiveShaft` determines the active transmission shaft and the active clutch, respectively. This information is very important in view to the control of the clutches, because the clutch control values are calculated independent of the clutch number. There is only one differentiation between the active and the inactive clutch, or during a shifting process between the engaging and the disengaging clutch. Doing this, each control value is calculated only once using an individual control block.

Currently, the clutch control distinguishes between creep, driveaway, drive, and shift. For all these drive modes exists a replaceable control. The controls calculate independent values for the active clutch. In consideration of the drive mode and the active shaft the block `ClutchXControl` assigns the control signal for the particular clutch. As shown in Figure 6, the clutch control signal is calculated using `Activate` and `ActivateOr` blocks with the state signals. Doing this, the independent control values of the different drive modes are merged to one control signal for the particular clutch. Finally, the transmission control contains the model `Synchronisation`, which controls the synchronisers of the particular gears. The control has to safeguard that only one gear per transmission shaft is synchronised.

The aim of the modelling and the simulation of a double clutch transmission are to get data of shifting processes of different quality. According to this, the focus of this work is on the control of the shifting process. The shifting process is controlled by the model `ControlShift`. There exist four different shifting types in consideration of downshift and upshift as well as pulling and pushing power of the power train. Additionally, these types are separated in two phases. The shifting types are: pull upshift, push downshift, pull downshift, and push upshift.

The structure of the model `ControlShift` is shown in Figure 7. For each shifting type and each phase exists a replaceable control using the same interface. The three output signals are *clutch_shift_old*, *clutch_shift_new*, and *engineEngagement*. The input is an activation signal in form of a Boolean signal. If the activation signal equals false, then the outputs have to be set to zero. Only one shifting process can be activated, because all blocks work in parallel. All types and phases are replaceable elements to simulate shifting processes of different quality.

The two frequently used shifting types are pull upshift and push downshift. They are similar in their control behaviour. In the first phase the torque is transmitted from the old clutch to the new clutch, so both clutches are engaged. Then in the second phase of the shifting process the engine speed is passed to the speed of the new shaft. The engine speed decreases while upshifting and increases while downshifting. Two possibilities exist for the adjustment of the engine speed. On the one hand it can be managed by an increase of the clutch capacity. Then the drive end impacts the engine torque and the engine speed decreases or increases, respectively. The clutch capacity is a function of the

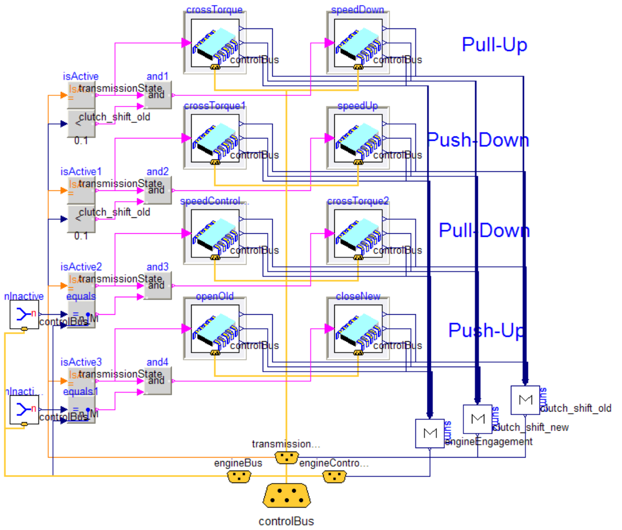

Figure 7: Shift control model with replaceable shift modes and phases

clutch engagement and describes the clutch's maximum possible transmission torque. The real transmitted torque of the clutch is a function of the clutch capacity and the input torque. Therefore, the clutch is either in slipping or locked mode. On the other hand it is possible to do an engine engagement. This means that the engine torque increases or decreases and to that effect the engine speed will be changed. In this work we use an engine based on an engine map. According to this, an engine engagement changes the internal throttle position as input of the engine map.

The other two shifting types are pull downshift and push upshift. Here, the order of the two phases changes. At first there is the adjustment of the engine speed to the second transmission shaft speed. Finally, the torque is transmitted from the first to the second clutch. The downshift with pulling power occurs in consequence of a kick down. Then, it is possible that both clutches are engaged during the speed adjustment, because of the demand of high power transmission. An upshift with pushing power occurs normally if the vehicle rolls downhill with increasing speed and the engine speed exceeds the upshift threshold.

Now, it is possible to simulate shifting processes of different quality. A good shifting results in an uninterrupted power transmission. In this case the driver does not sense a changing of the acceleration. Then the sum of both clutch torques equals the engine torque less the dynamic torques at any time. A bad shifting in consequence of control errors can result in revolution speed droppings, break outs, or oscillations. This leads to a changing of the acceleration and will be sensed by the driver.

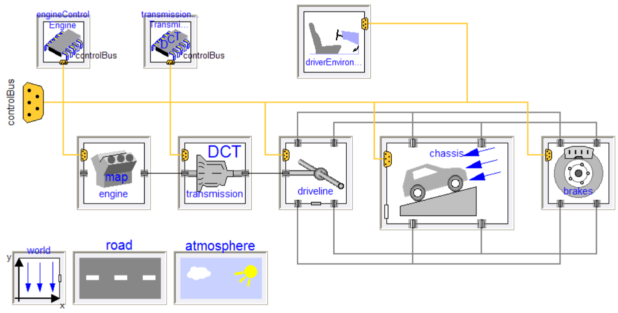

Figure 8: Vehicle model with replaceable elements

5 Simulation of shifting cycles

With the described models and controls, simulations of different shifting cycles are executed. By varying the driver models it is possible to drive miscellaneous drive and shifting cycles, e. g., in form of a table with acceleration and brake pedal positions, or following a velocity profile in form of predefined drive cycles. An example for a drive cycle is the New European Driving Cycle NEDC [6].

The vehicle simulation structure is shown in Figure 8. The components are replaceable or have internal replaceable elements. Furthermore, it is possible to use existing models, e. g., of the *PowerTrain* library. Both the redeclaration of physical components and the shift control can be changed. Thus, the simulations generate data of different quality due to the choice of the grade of the control.

6 Results and application

Based on the presented vehicle simulation structure we can now simulate drive and shifting cycles. As an example, the simulation results of a drive cycle are shown in Figure 9. The drive cycle contains the following sections: engine start, creep, driveaway, upshifts, drive, downshifts, and stoppage. The upshifts are realised with an engine engagement and the downshifts are realised with an increase of the clutch capacity for the adjustment of the engine speed.

At the top the Figure 9 shows the speeds of the engine and of the two transmission shafts. In the next part the capacity of the two clutches and the engine torque can be seen. The up- and downshifts can be easily recognised by the crossing of the clutch capacities and the change of the engine speed between the

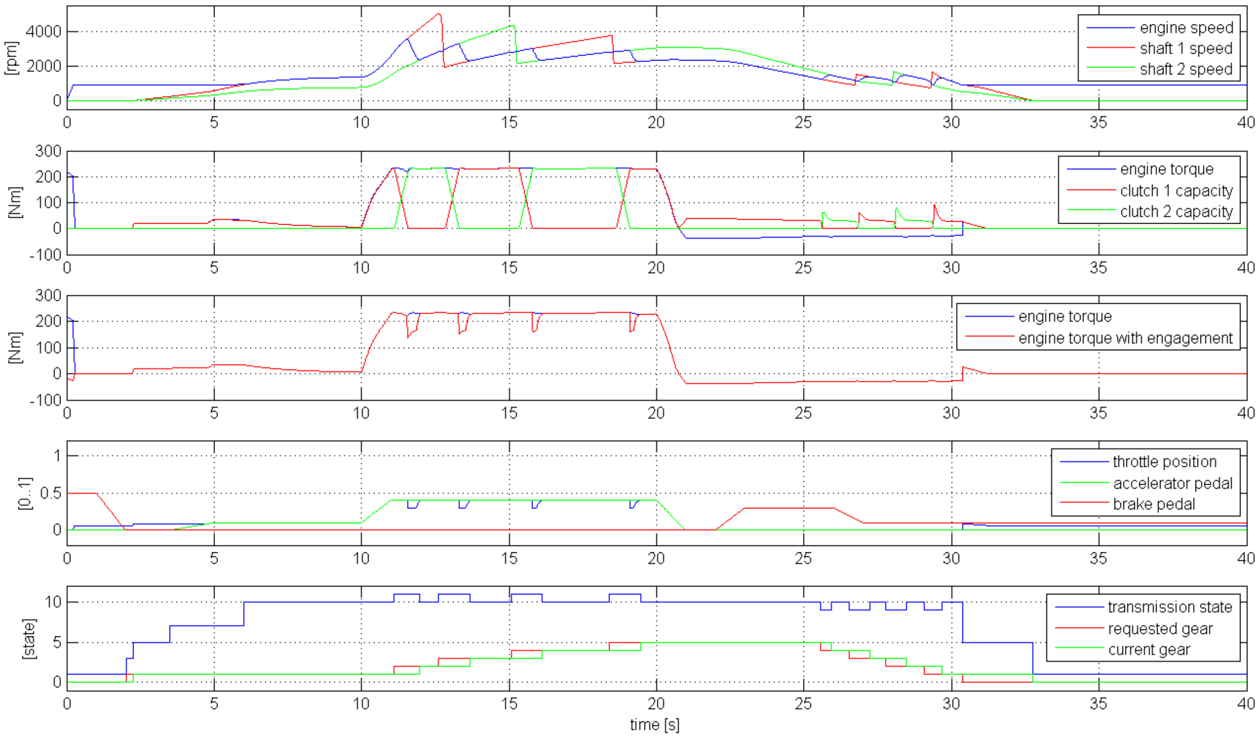

Figure 9: Simulation results of a simple drive cycle

transmission shaft speeds. During the downshifts the clutch capacity increases for the adjustment of the engine speed as described in Section 4. The third part shows the engine engagement at the upshifts. At the upshift the engine torque is decreased during the engine speed adjustment to the new transmission shaft speed. The engine control decreases the throttle position on demand of the transmission control to decrease the engine torque. The acceleration pedal, the brake pedal, and the decreased throttle position can be seen in the fourth plot of Figure 9. The throttle position increases at the start and at the end of the simulation, because of the engagement of the engine speed governor. The bottom of Figure 9 shows the transmission state according to the described sections. Furthermore, the current and the requested gear can be seen. These three signals are state signals and the Integer value represents a particular state.

In the past, we developed in cooperation with the IAV GmbH an automated measurement data analysis system [1]. The evaluation process of the software can be parameterised efficiently by XML templates. The system supports the common automotive measurement file formats and it can handle huge data traces with a sequential data processing. Moreover, there exist several intelligent signal processing modules for the data evaluation.

For complex analyses the system parametrisation is difficult, because the measurement data is often unlabelled . The data contains measurements of several shifting processes, but there is normally no information about good or bad examples. Now, we have the possibility to simulate shifting processes of different quality. With the simulation results of this work we can parameterise the automated measurement data analysis system for the evaluation of shifting processes. The aim of the measurement data analysis is the detection and assessment of bad shifting in the vehicle development.

7 Conclusion and outlook

In this paper, we presented the modelling, control, and simulation of a double clutch transmission. The focus is the simulation of data of different quality particularly with regard to shifting processes. According to this, we developed model and control structures with replaceable elements. It is possible to use the same vehicle model for the simulation of good and bad shiftings. On the one hand the level of detail can be changed. On the other hand it is possible to redeclare elements of the transmission control to change the control behaviour.

We presented a library with basic transmission components and introduced state signals to access sys-

tem states from different points of the control process. Furthermore, we described the developed transmission control and discussed the different shifting types in detail. The vehicle model can be used to simulate a power train with a double clutch transmission. Finally, the simulation results of a simple drive cycle were shown.
With the presented models we get the background for future works. We developed a base model of a double clutch transmission and a base control structure with a working control for the four basic shifting types. At the moment there exist basic controls for each phase of the shifting. The next step will be the variation of the control to get data of different quality. Additionally, we will simulate drive cycles using double clutch transmission models of different levels of detail. With these results we will parameterise the automated measurement data analysis system for the evaluation of shifting processes of double clutch transmissions in real vehicle measurement data.

References

[1] Henrik Isernhagen, Helmut Neemann, Steffen Kühn, and Clemens Gühmann. Intelligent signal processing in an automated measurement data analysis system. In *Proceedings of the IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)*, pages 83 – 87, Honolulu, USA, April 2007.

[2] Mike Dempsey, Magnus Gäfvert, Peter Harman, Christian Kral, Martin Otter, and Peter Treffinger. Coordinated automotive libraries for vehicle system modelling. In *Proceedings of the 5th International Modelica Conference*, pages 33–41, Vienna, September 2006. The Modelica Association and arsenal research.

[3] Jakub Tobolář, Martin Otter, and Tilman Bünte. Modelling of Vehicle Powertrains with the Modelica PowerTrain Library. In *Systemanalyse in der Kfz-Antriebstechnik IV*, volume 79 of *Haus der Technik Fachbuch*, pages 204–216, Augsburg, February 2007.

[4] Modelica. http://www.modelica.org.

[5] M. Otter, K.-E. Årzén, and I. Dressler. StateGraph - A Modelica Library for Hierarchical State Machines. In *Proceedings of the 4th International Modelica Conference*, pages 569–578, Hamburg, March 2005. The Modelica Association and the Hamburg University of Technology.

[6] Emission Test Cycles. Summary of worldwide engine and vehicle test cycles. http://www.dieselnet.com/standards/cycles/.